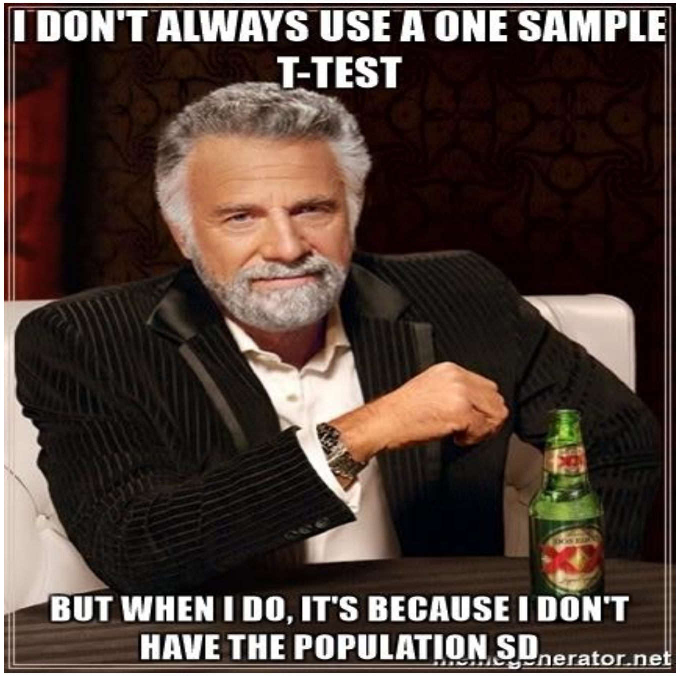
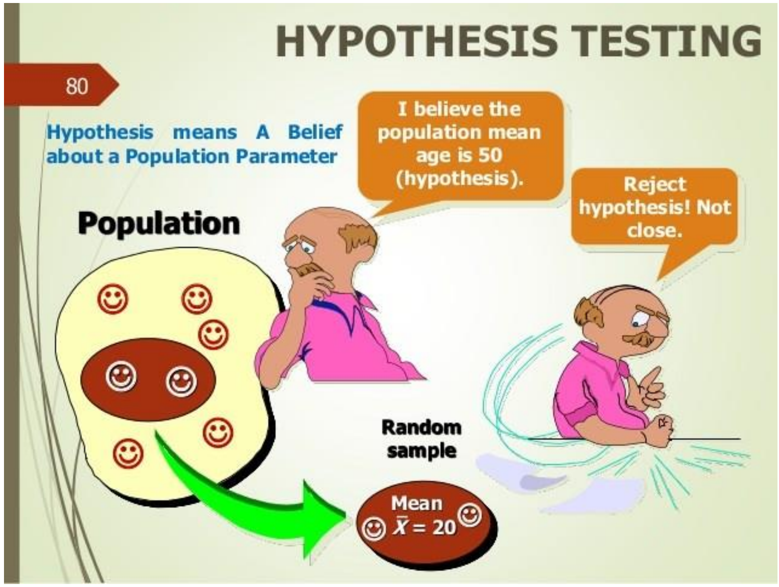
# Hypothesis Testing in Python


I DON'T ALWAYS USE A ONE SAMPLE T-TEST
BUT WHEN I DO, IT'S BECAUSE I DON'T HAVE THE POPULATION SD


HYPOTHESIS TESTING
80
Hypothesis means A Belief about a Population Parameter
I believe the population mean age is 50 (hypothesis).
Reject hypothesis! Not close.
Population
Random sample
Mean $\bar{X}$ = 20

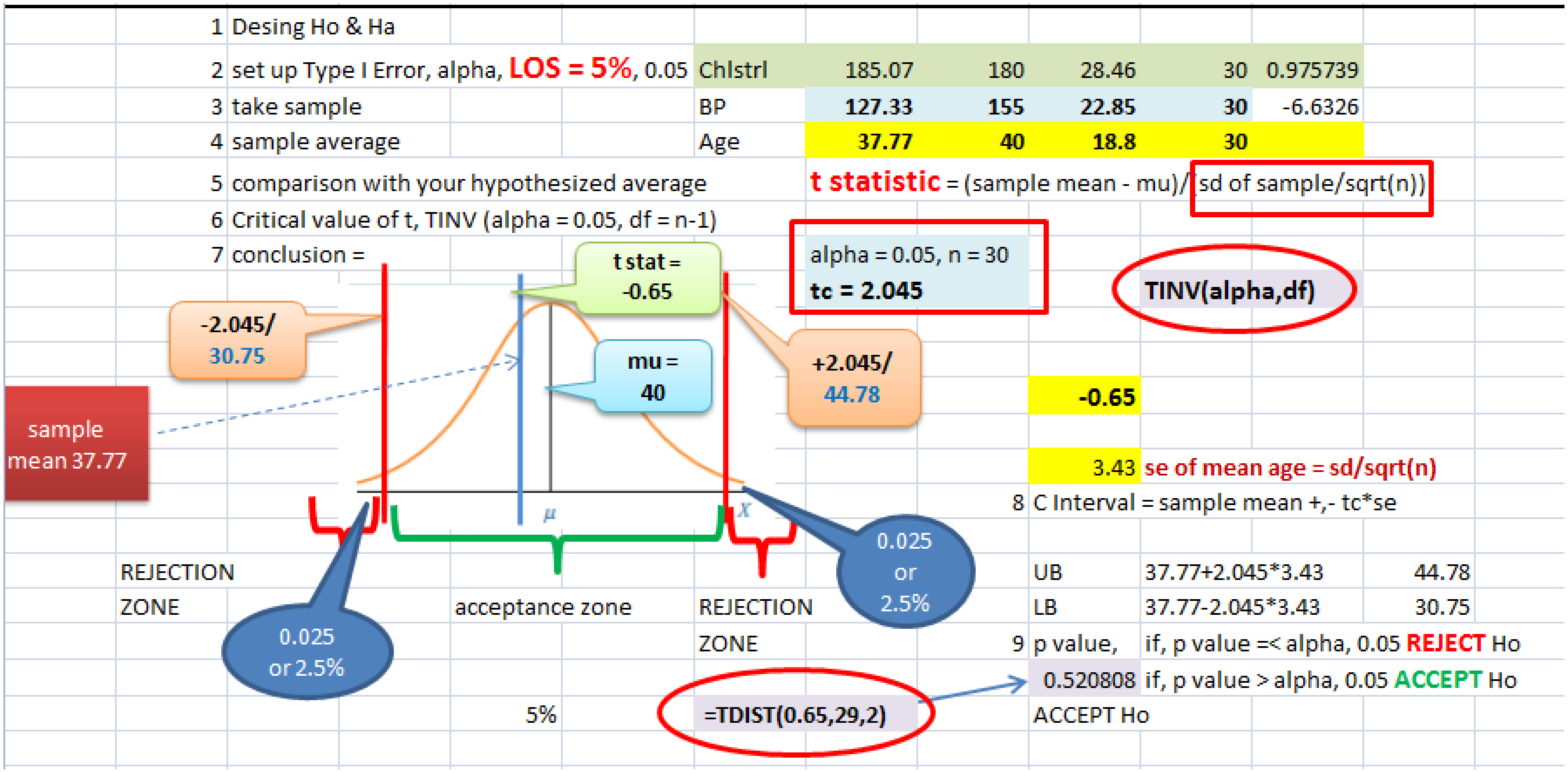
# One Sample *t* test

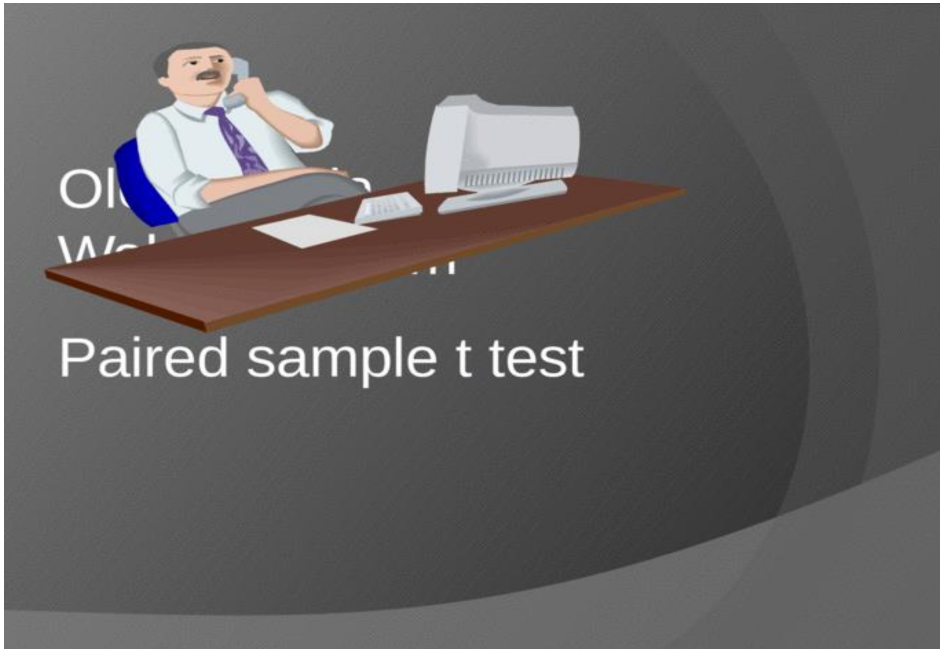
Ho: Hypothesized Mean population *Age,* mu = 40

```
>>> import scipy
>>> onesam = scipy.stats.ttest_1samp(a = cs2m.Age,
popmean = 40)
>>> print(onesam)
Ttest_1sampResult(statistic=-0.65080281293680164,
pvalue=0.52029737629801631)
```

```
> t.test(cs2m$Age, mu = 40)

        One Sample t-test

data:  cs2m$Age
t = -0.6508, df = 29, p-value = 0.5203
alternative hypothesis: true mean is not equal to 40
95 percent confidence interval:
 30.74814 44.78520
sample estimates:
mean of x
 37.76667
```

# Ho: Hypothesized Mean population *Age,* mu = 40




Paired sample t test

# Paired sample:

Ho: Mu of *quiz1*- Mu of *quiz2* = 0

```
>>> pairedsam = scipy.stats.ttest_rel(
grades.quiz1, grades.quiz2)
>>> print(pairedsam)
Ttest_relResult(statistic=-2.871706119
2333544, pvalue=0.0049483120272184486)
>>>
```

Python output is same as that of R

```
> t.test(grades$quiz1, grades$quiz2, paired = T)

        Paired t-test

data:  grades$quiz1 and grades$quiz2
t = -2.8717, df = 104, p-value = 0.004948
alternative hypothesis: true difference in means is
not equal to 0
95 percent confidence interval:
 -0.8694223 -0.1591491
sample estimates:
mean of the differences
             -0.5142857
```

# Paired sample:

## Ho: Mu of *quiz1*- Mu of *quiz2* = 0

| Quiz1 | Quiz2 | d | $d^2$ |
|---|---|---|---|
| 6 | 5 | 1 | 1 |
| 10 | 10 | 0 | 0 |
| 10 | 7 | 3 | 9 |
| 7 | 8 | -1 | 1 |
| 10 | 9 | 1 | 1 |
| 7 | 8 | -1 | 1 |
| | Total= | -54 | 378 |
| | Average= | -0.51429 | |

$$Confidence\ Interval = Mean \pm 1.96 \times Standard\ Error\ of\ Mean\ (Difference)$$

| | | |
|---|---|---|
| Upper Limit | -0.163274866 | Zc |
| Lower limit | -0.865296562 | |

| | t critical |
|---|---|
| =TINV(0.05,104) | 1.983037471 |
| LB | -0.869422278 |
| UB | -0.159149151 |

$$s_d = \sqrt{\frac{\sum(d^2)}{n-1} - \frac{(\sum d)^2}{n(n-1)}}$$

| | |
|---|---|
| sigma d^2/(n-1) | 3.634615 |
| (sigma d)^2/(n(n-1)= | 0.267033 |
| Diff, left & rt expressio | 3.367582 |
| Square root, SD = | 1.835097 |
| SE of mean = | 0.17909 |

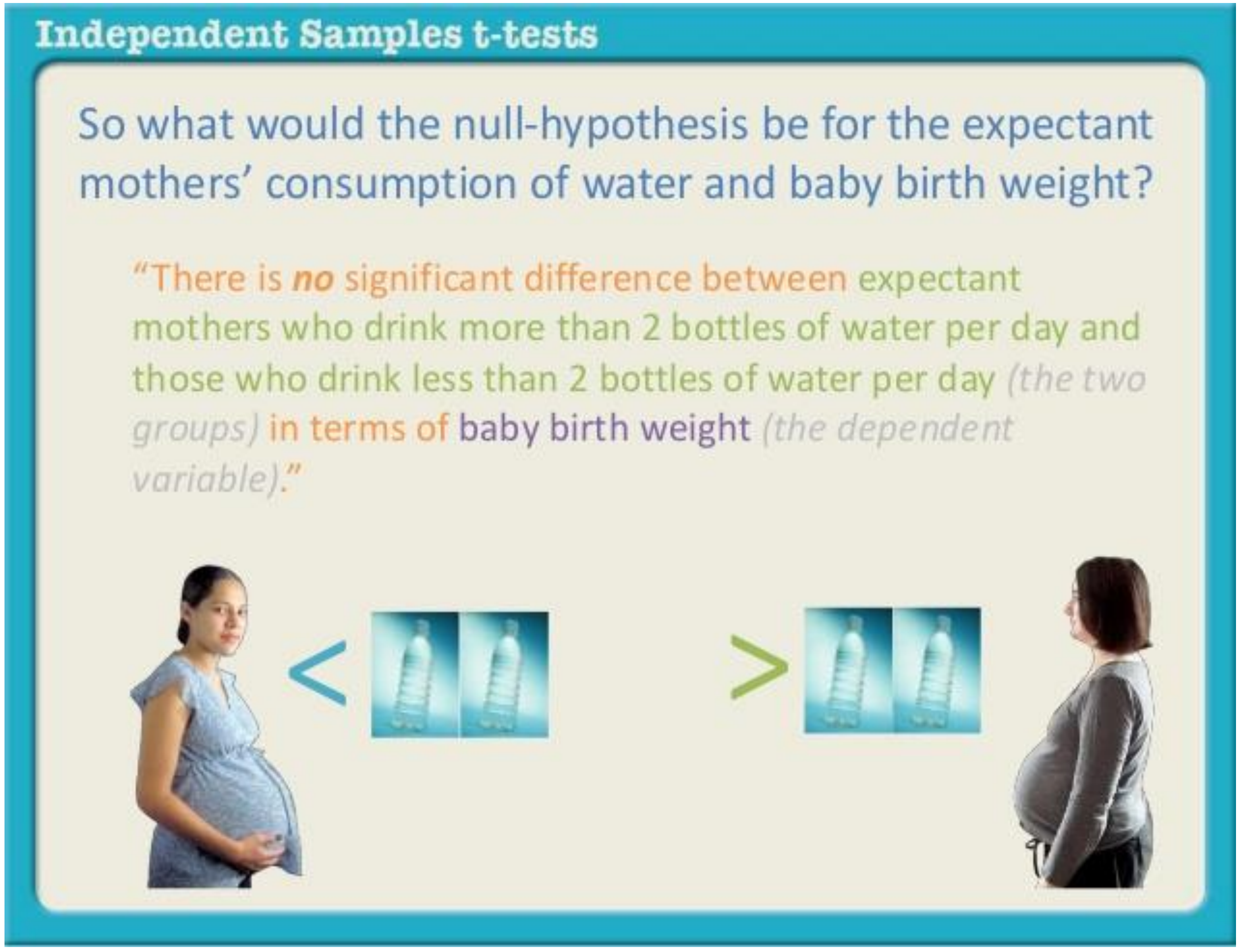
$$t - calc = \frac{\bar{d} - \mu_d}{s_d/\sqrt{n}} = \frac{-0.514 - 0}{1.8351/\sqrt{105}} = -2.872$$

```
> t.test(grades$quiz1, grades$quiz2, paired = T)

        Paired t-test

data:  grades$quiz1 and grades$quiz2
t = -2.8717, df = 104, p-value = 0.004948
alternative hypothesis: true difference in means is
not equal to 0
95 percent confidence interval:
 -0.8694223 -0.1591491
sample estimates:
mean of the differences
             -0.5142857
```

| | |
|---|---|
| 0.00494406 | =TDIST(2.872,104,2) |

**Python output is same as that of R**

```
>>> pairedsam = scipy.stats.ttest_rel(
grades.quiz1, grades.quiz2)
>>> print(pairedsam)
Ttest_relResult(statistic=-2.871706119
2333544, pvalue=0.004948312027218486)
```


Independent Samples t-tests
So what would the null-hypothesis be for the expectant mothers' consumption of water and baby birth weight?
"There is *no* significant difference between expectant mothers who drink more than 2 bottles of water per day and those who drink less than 2 bottles of water per day *(the two groups)* in terms of baby birth weight *(the dependent variable)*."
<
>

# Independent Sample *t* test

## Ho: Population mean of *BP* across *Anxiety* levels are same

```
>>> cs2m.shape
(30, 6)
>>> cs2m_AnxtyL = cs2m[cs2m.AnxtyLH == 0]
>>> cs2m_AnxtyL.shape
(16, 6)
>>> cs2m_AnxtyH = cs2m[cs2m.AnxtyLH == 1]
>>> cs2m_AnxtyH.shape
(14, 6)
>>> import scipy
>>> scipy.stats.ttest_ind(cs2m_AnxtyL.BP, cs2m_AnxtyH.BP)
Ttest_indResult(statistic=-2.6896732510162993, pvalue=0.011916830524990
729)
```

Create a data set having LOW Anxiety

Create a data set having HIGH Anxiety

Python output is same as that of R

```
> t.test(cs2m$BP~cs2m$AnxtyLH, var.equal = TRUE)

        Two Sample t-test

data:  cs2m$BP by cs2m$AnxtyLH
t = -2.6897, df = 28, p-value = 0.01192
alternative hypothesis: true difference in means is not
 equal to 0
95 percent confidence interval:
 -35.93942  -4.86415
sample estimates:
mean in group 0 mean in group 1
       117.8125        138.2143
```


# *Tip*

*Independent sample t test is not that straight forward as in R. Need to create a data set having one chosen category only, say Pregnant like this:*

***Prgnt = cs2m[cs2m.Prgnt == 1]***
***NotPrgnt = cs2m[cs2m.Prgnt == 0]***

*Then run the code to see the mean population difference in BP as:*

***scipy.stats.ttest_ind(Prgnt.BP, NotPrgnt.BP)***

# Independent Sample *t* test

## Ho: Population mean of *BP* across *Anxiety* levels are same

| BP | Anxty | BP | Anxty |
|---|---|---|---|
| 100 | 0 | 110 | 1 |
| 120 | 0 | 120 | 1 |
| 110 | 0 | 150 | 1 |
| 100 | 0 | 160 | 1 |
| 95 | 0 | 120 | 1 |
| 125 | 0 | 120 | 1 |
| 135 | 0 | 125 | 1 |
| 165 | 0 | 115 | 1 |
| 145 | 0 | 150 | 1 |
| 120 | 0 | 130 | 1 |
| 100 | 0 | 170 | 1 |
| 100 | 0 | 145 | 1 |
| 95 | 0 | 180 | 1 |
| 120 | 0 | 140 | 1 |
| 125 | 0 | | |
| 130 | 0 | | |
| 117.81 | <-avg-> | 138.21 | |

| | | | | | |
|---|---|---|---|---|---|
| Mean x1= | 117.81 | n1= | 16 | s1= | 19.83 |
| Mean x2= | 138.21 | n2= | 14 | s2= | 21.46 |

σ=Spooled

| | |
|---|---|
| Numerator= | 11885 |
| Denom= | 28 |
| σ=(N/D)^0.5= | 20.603 |

$$S_{pooled} = \sqrt{\frac{s_1^2(n_1-1)+s_2^2(n_2-1)}{n_1+n_2-2}}$$

*t* calc

| | |
|---|---|
| Num= | -20.4 |
| Denom=SE= | 7.5399 |
| *t* calc | -2.706 |

$$t-calc = \frac{(\bar{x}_1-\bar{x}_2)-(\mu_1-\mu_2)}{\sigma\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}$$

| | |
|---|---|
| 0.008 | =TDIST(2.706,103,2) |

```
>>> import scipy
>>> scipy.stats.ttest_ind(cs2m_AnxtyL.BP, cs2m_AnxtyH.BP)
Ttest_indResult(statistic=-2.6896732510162993, pvalue=0.011916830524990
729)
```

```
> t.test(cs2m$BP~cs2m$AnxtyLH, var.equal = TRUE)

        Two Sample t-test

data:  cs2m$BP by cs2m$AnxtyLH
t = -2.6897, df = 28, p-value = 0.01192
alternative hypothesis: true difference in means is not
 equal to 0
95 percent confidence interval:
 -35.93942  -4.86415
sample estimates:
mean in group 0 mean in group 1
       117.8125        138.2143
```


Analysis of Varia
(ANOVA)
One Way
Classification

# One Way ANOVA

## Ho: *sales* across *city* is same

```
>>> import statsmodels.api as sm
C:\Users\iNurture\Anaconda3\lib\site-packages\statsmodels\compat\pandas.py:56
: FutureWarning: The pandas.core.datetools module is deprecated and will be r
emoved in a future version. Please use the pandas.tseries module instead.
  from pandas.core import datetools
>>> from statsmodels.formula.api import ols
```

```
>>> mod = ols('sales~city', data = salescity).fit()
>>> aov_table = sm.stats.anova_lm(mod, type = 2)
>>> print(aov_table)
            df     sum_sq    mean_sq          F        PR(>F)
city       3.0  59.708333  19.902778  43.033033  6.539131e-09
Residual  20.0   9.250000   0.462500        NaN           NaN
>>>
```

# One Way ANOVA

## Ho: *sales* across *city* is same

```
>>> mod = ols('sales~city', data = salescity).fit()
>>> aov_table = sm.stats.anova_lm(mod, type = 2)
>>> print(aov_table)
            df      sum_sq    mean_sq          F        PR(>F)
city       3.0   59.708333  19.902778  43.033033  6.539131e-09
Residual  20.0    9.250000   0.462500        NaN           NaN
>>>
```

**Python output is same as that of R**

```
> result<- aov(sales~city, data = salescity)
> summary(result)
            Df Sum Sq Mean Sq F value   Pr(>F)    
city         3  59.71  19.903   43.03 6.54e-09 ***
Residuals   20   9.25   0.462                     
---
Signif. codes:  
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

# One Way ANOVA

# Ho: *sales* across *city* is same

| | $j_{th}$ column | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| Stores | ***Delhi*** | ***Mumbai*** | ***Kolkata*** | ***Chennai*** | |
| 1 | 22 | 19 | 18 | 21 | |
| 2 | 22.5 | 19.5 | 17 | 20 | |
| 3 | 21.5 | 19 | 18.5 | 21.5 | |
| 4 | 22 | 20 | 17 | 20 | |
| 5 | 22.5 | 19 | 18.5 | 21 | |
| 6 | 21.5 | 21 | 17 | 20 | |
| Total, T= | 132 | 117.5 | 106 | 123.5 | 479 |
| Avg = | **22** | **19.58** | **17.67** | **20.58** | **19.96** |

$$Sum\ Square\ between\ colums = \sum_{j=1}^{k}\left[n_j\left(\bar{x}_j - \bar{\bar{x}}\right)^2\right]$$

SSC= **59.70833333**

$$SSE = Sum\ Squares\ within\ Samples = \sum_{i=1}^{n}\sum_{j=1}^{k}\left(x_{ij} - \bar{x}_j\right)^2$$

SSE = **9.25**

$$SST = Total\ Sum\ Squaress = \sum_{i=1}^{n}\sum_{j=1}^{k}\left(x_{ij} - \bar{\bar{x}}\right)^2$$

SST= **68.87152778**

$$MSC\ mean\ square\ column = \frac{SSC}{k-1}$$

**MSC= 19.9028**

$$MSE\ mean\ square\ within\ samples = \frac{SSE}{n-k}$$

**MSE= 0.4625**

$$F - statistics = \frac{MSC}{MSE}$$

**F-stat= 43.033**

**Ho: Mean sales across cities is same**

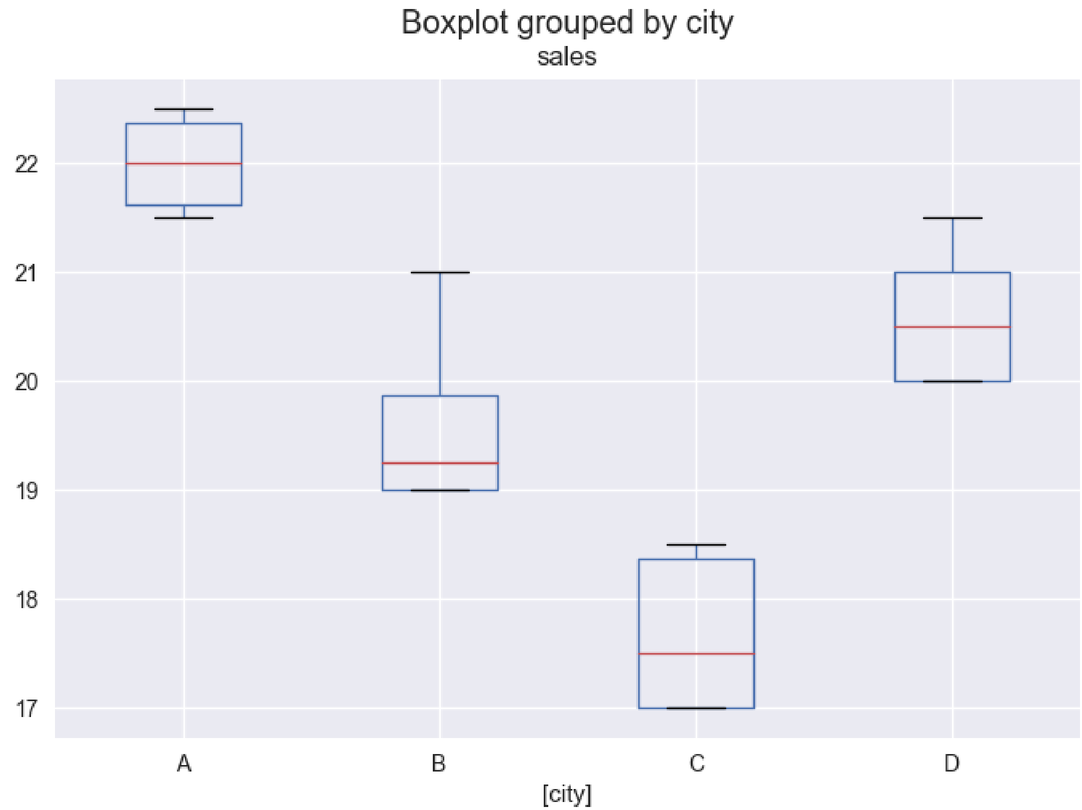
**Ha: at least one city's sale is different from others**

```
> results<- aov(sales~city, data = salescity)
>
> summary(results)
            Df Sum Sq Mean Sq F value   Pr(>F)
city         3  59.71  19.903   43.03 6.54e-09 ***
Residuals   20   9.25   0.462
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

```
>>> mod = ols('sales~city', data = salescity).fit()
>>> aov_table = sm.stats.anova_lm(mod, type = 2)
>>> print(aov_table)
            df     sum_sq    mean_sq          F        PR(>F)
city       3.0  59.708333  19.902778  43.033033  6.539131e-09
Residual  20.0   9.250000   0.462500        NaN           NaN
>>>
```

# Boxplots of *Sales* vs *City*

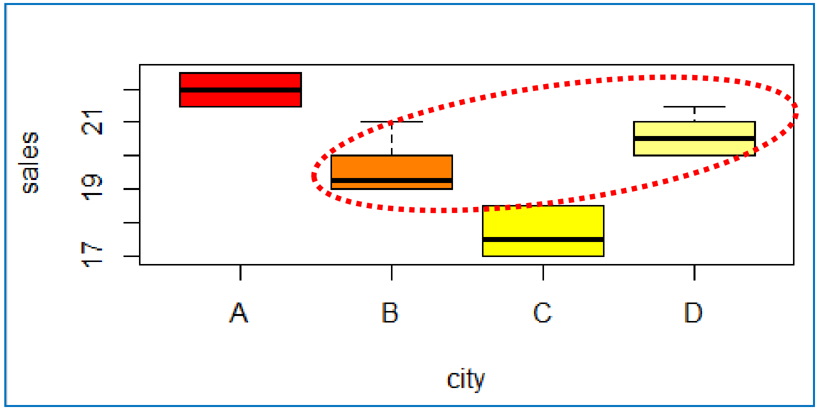
```
>>> salescity.boxplot(by = 'city')
```



# Tukey HSD

```
>>> from statsmodels.stats.multicomp import pairwise_tukeyhsd
>>> tukey = pairwise_tukeyhsd(salescity.sales, salescity.city, alpha=0.05)
>>> print (tukey)
Multiple Comparison of Means - Tukey HSD,FWER=0.05
=============================================
group1 group2 meandiff  lower   upper  reject
---------------------------------------------
  A      B    -2.4167  -3.5157 -1.3176  True
  A      C    -4.3333  -5.4324 -3.2343  True
  A      D    -1.4167  -2.5157 -0.3176  True
  B      C    -1.9167  -3.0157 -0.8176  True
  B      D      1.0     -0.099  2.099  False
  C      D     2.9167   1.8176  4.0157  True
---------------------------------------------
```

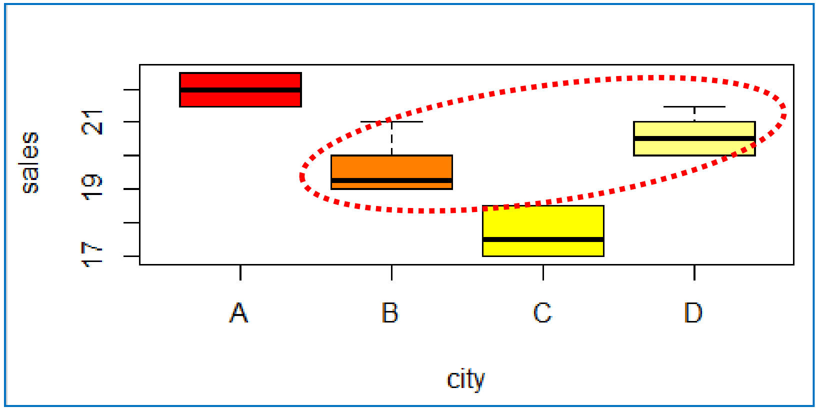
Alternate Hypothesis




**H0: Mean sales across city B and D are same**

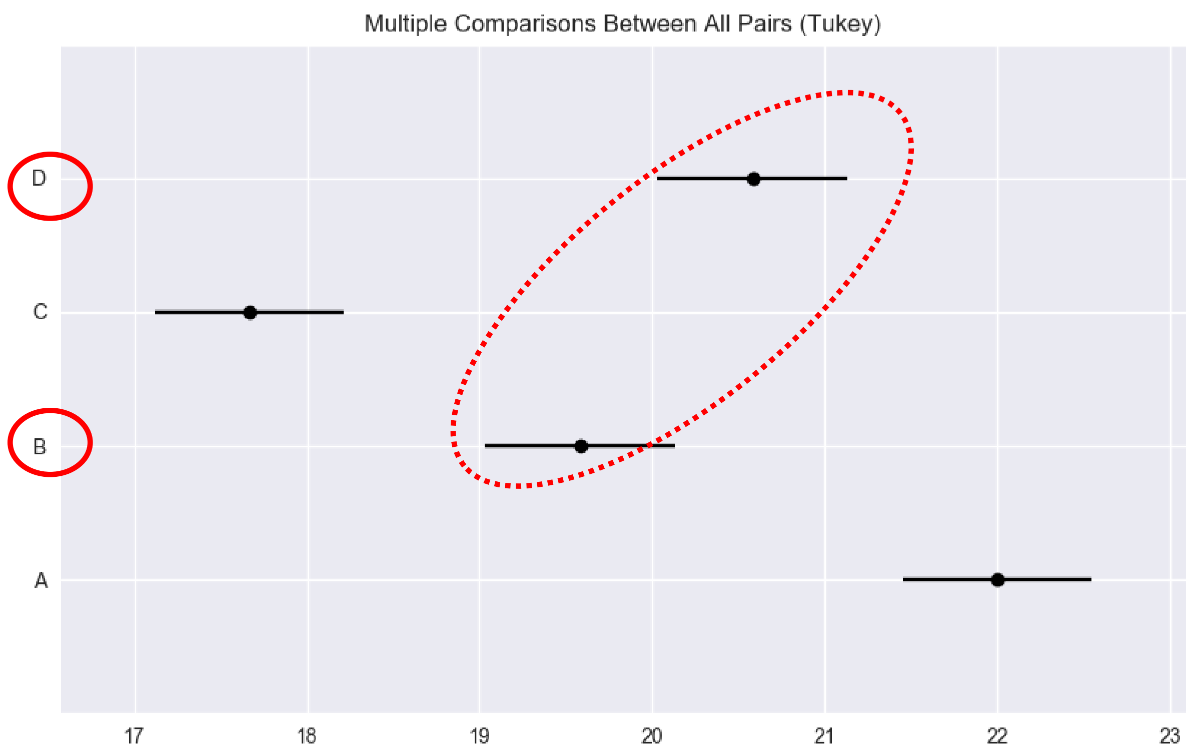
# Tukey Result

```
> TukeyHSD(result, conf.level = 0.95)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = sales ~ city, data = salescity)

$city
         diff         lwr        upr     p adj
B-A -2.416667 -3.51564273 -1.3176906 0.0000286
C-A -4.333333 -5.43230939 -3.2343573 0.0000000
D-A -1.416667 -2.51564273 -0.3176906 0.0087518
C-B -1.916667 -3.01564273 -0.8176906 0.0004849
D-B  1.000000 -0.09897606  2.0989761 0.0826671
D-C  2.916667  1.81769061  4.0156427 0.0000020
```



# Tukey Plot

```
>>> tukey.plot_simultaneous()
```



# CHI SQUARED TEST

# Chi Square Test

Ho: There is no significant association between *Anxiety* and *Drug Reaction*

```
>>> import pandas as pd
>>> pd.crosstab(cs2m.AnxtyLH, cs2m.DrugR, margins = True)
DrugR     0   1  All
AnxtyLH
0        11   5   16
1         4  10   14
All      15  15   30
>>> AnxtyDrug = np.array([[11,5],[4,10]])
>>> scipy.stats.chi2_contingency(AnxtyDrug)
(3.3482142857142856, 0.067277960538349058, 1, array([[ 8.,  8.],
       [ 7.,  7.]]))
```

First, do crosstab then create array then use array in test


# Tip

*Chi square test is not that straight forward as in R. Need to create cross tab of the two Categorical variables like:*

***pd.crosstab(cs2m$Prgnt, cs2m$AnxtyLH, margins = True)***

*Then create an array having observed frequencies/counts like:*

***PrgntAnxty = np.array([[x1,x2], [x3,x4]])***

*Now, run the code as:*

***stats.chi2_contingency(PrgntAnxty)***

Be careful in feeding x1, x2, x3 & x4. Refer example of Anxiety versus Drug Reaction

# Chi Square Test

Ho: There is no significant association between *Anxiety* and *Drug Reaction*

**Same in R & Python**

```
> chisq.test(cs2m$AnxtyLH, cs2m$DrugR)

        Pearson's Chi-squared test with Yates' continuity
        correction

data:  cs2m$AnxtyLH and cs2m$DrugR
X-squared = 3.3482, df = 1, p-value = 0.06728
```

```
>>> import pandas as pd
>>> pd.crosstab(cs2m.AnxtyLH, cs2m.DrugR, margins = True)
DrugR     0   1  All
AnxtyLH
0        11   5   16
1         4  10   14
All      15  15   30
>>> AnxtyDrug = np.array([[11,5],[4,10]])
>>> scipy.stats.chi2_contingency(AnxtyDrug)
(3.3482142857142856, 0.067277960538349058, 1, array([[ 8.,  8.],
       [ 7.,  7.]]))
```

First, do crosstab then create array then use array in test

**Same in Python & R**

# Chi Square Test

Ho: There is no significant association between *Anxiety* and *Drug Reaction*

**AnxtyLH * DrgR Crosstabulation**

| | | DrgR 0 | DrgR 1 | Total |
|---|---|---|---|---|
| AnxtyLH | 0 | 11 | 5 | 16 |
| | 1 | 4 | 10 | 14 |
| | Total | 15 | 15 | 30 |

Degrees of Freedom=
v = (R-1)*(C-1)
v = (2-1)*(2-1) = **1**

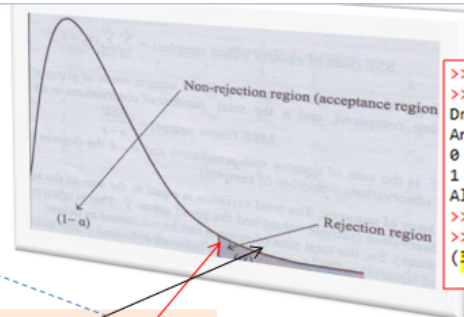
| | | DrgR 0 | | DrgR 1 | | |
|---|---|---|---|---|---|---|
| | | Obs | *Exp* | Obs | *Exp* | Row Total |
| AnxtyLH | 0 | 11 | *8* | 5 | *8* | 16 |
| | 1 | 4 | *7* | 10 | *7* | 14 |
| | Column Total | 15 | | 15 | | 30 |

Chi-Square CRITICAL VALUE
0.05 LOS & df 1 = 3.8415

| Obs | Exp | $(O-E)^2$ | $(O-E)^2/E$ |
|---|---|---|---|
| 11 | 8 | 9 | 1.125 |
| 4 | 7 | 9 | 1.285714 |
| 5 | 8 | 9 | 1.125 |
| 10 | 7 | 9 | 1.285714 |
| $\sum(O-E)^2/E$ | | | **4.82143** |

$$Expected\ Counts\ for\ cell = \frac{Row\ Total \times Column\ Total}{Grand\ Total}$$

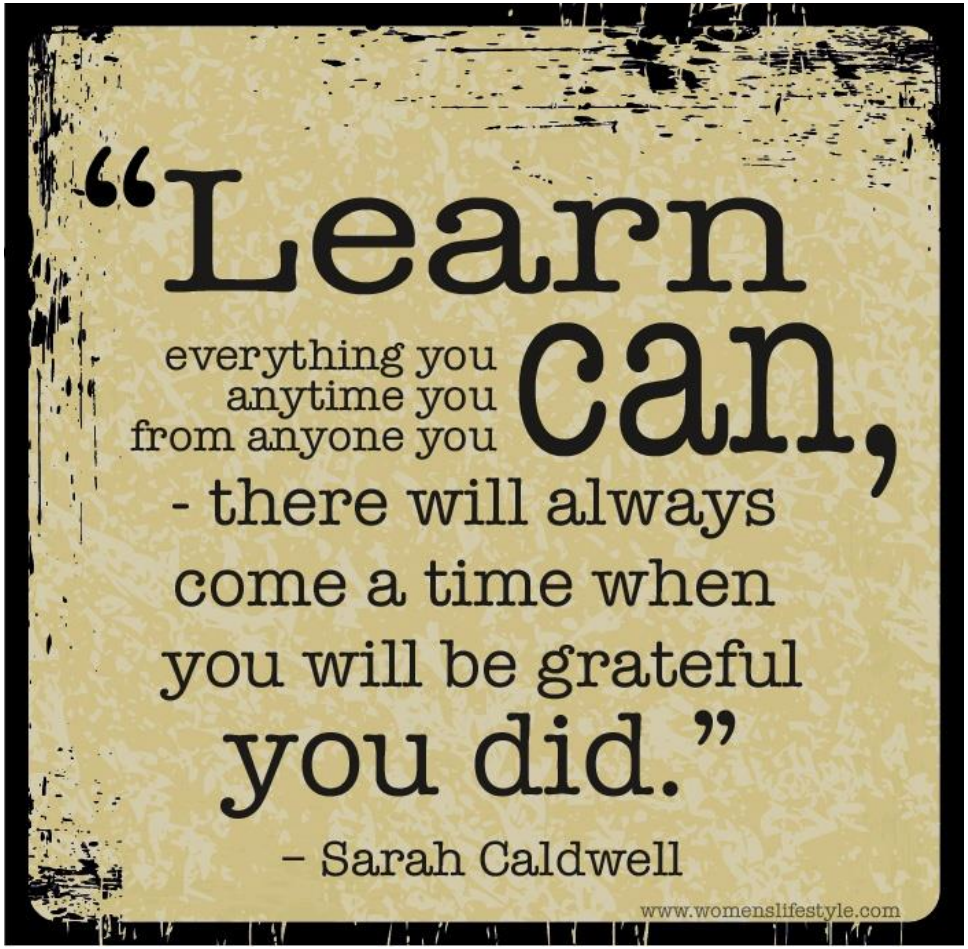
$$Chi-square\ Statistics = \sum \frac{(Obs - Exp)^2}{Exp}$$




```
>>> import pandas as pd
>>> pd.crosstab(cs2m.AnxtyLH, cs2m.DrugR, margins = True)
DrugR      0   1  All
AnxtyLH
0         11   5   16
1          4  10   14
All       15  15   30
>>> AnxtyDrug = np.array([[11,5],[4,10]])
>>> scipy.stats.chi2_contingency(AnxtyDrug)
(3.348214285714285‌6, 0.06727796053834905‌8, 1, array([[ 8.,  8.],
       [ 7.,  7.]]))
```

```
> chisq.test(cs2m$AnxtyLH, cs2m$DrugR)

        Pearson's Chi-squared test with Yates' continuity
        correction

data:  cs2m$AnxtyLH and cs2m$DrugR
X-squared = 3.3482, df = 1, p-value = 0.06728
```

| | NEATNESS | OCCUPATION |
|---|---|---|
| | 0=not neat, 1 neat | 0=homemaker, 1=working |
| 1 | 0 | 0 |
| 2 | 1 | 1 |
| 3 | 0 | 1 |
| 50 | 1 | 0 |


"Learn
everything you
anytime you
from anyone you
can,
- there will always
come a time when
you will be grateful
you did."
– Sarah Caldwell
www.womenslifestyle.com